Web Interface Program Design

The design of the web application is restricted by the web technologies used. It is possible to find this or that work around but this is never a general solution. Using another browser or different versions of the same browser can have significant effects. Setting up browser security to a different level can change the application's environment.

This section covers the following topics:

- Data Input
- Global Data
- Dispatch
- External Data
- Modular Pages
- Choosen Program Design

Data Input

All data given by the HTTP server and web browser must be treated as unchecked data! Nobody can force a user to use scripting, given colors or other settings.

The application has to check all input data again for consistency, even if the calling page contains scripts for validation. Not only the value of the given data is unchecked, even the data format and length is insecure. An HTML page always can be copied with "Save as" and then modified for your own needs.

Use the Natural IS Option for checking format and length of values. The example below shows reading of the value START followed by checking whether the value is an N5 value.

```
PERFORM W3READ-ENVIRONMENT "START" "P" W3VALUE W3MAX
 IF W3MAX GT 0 THEN
   IF W3VALUE-A5 IS (N5) THEN
      #DEMO-PARM.#START:= VAL(W3VALUE-A5)
   ELSE
     #DEMO-PARM.#START:= 1
   END-IF
 ELSE
   #DEMO-PARM.#START:= 1
 END-IF
```

Global Data

With your web application, you have different ways of saving global data.

1. Cookies

Cookies are data saved with your local browser. The number of cookies is limited. Use only one cookie for your whole application, to save different data needed for your application.

Cookies are delivered with the variable HTTP_COOKIE. The Natural Web Server Extension used has to be advised to pass cookies to your application. Your initialization file must contain the following line:


```
ENV=HTTP_COOKIE
```

Inside your program, cookies can be read with the normal functionality:

```
PERFORM W3READ-ENVIRONMENT "HTTP_COOKIE" 'S' W3VALUE W3MAX
IF W3MAX > 0 THEN
 #MY-COOKIE := W3VALUE
ELSE
 RESET #MY-COOKIE
END-IF
```

Add your cookie, which has to be set with an expire time, at the return page:

```
ADD 1 TO #NUMBER
COMPRESS "COUNTER=" #NUMBER ";" INTO COOKIE-NAME LEAVING NO
COMPRESS 'Set-Cookie:' COOKIE-NAME
  'expires=Wednesday, 09-Nov-99 23:12:40 GMT' ##HTTP_NEWLINE_END
  INTO W3VALUE
PERFORM W3HTTP W3VALUE
```

2. Hidden Input Fields and Additional URL Parameters

If your page contains a form, additional data can be saved as a hidden input field. This data will be sent if the form is submitted, but cannot be seen when the HTML page is displayed. If not form tag is used, it is possible to add this parameter direct to the URL of the next pages called.

Setting of "global data" using hidden input fields inside a form:

```
PERFORM H3-INPUT "H" "START" #START 0 0
PERFORM H3-INPUT "H" "FROM"  #FROM  0 0
PERFORM H3-INPUT "H" "TO"    #TO    0 0
```

3. Data Saved on the Server Side (in a Database)

It is very common to save data for a given user at the application database. This is only useful if password saved applications are used. The username then can be used to load specific data out of the database.

Dispatch

For many applications, it is common that for a given input, different actions can be called. However, submitting a form tag can only call one program.

One way of solving the problem is using scripting and changing the URL of the called program depending on the submit button pressed.

A second way is to create one dispatcher program that evaluates which program should be called, depending on the pressed submit button.

A combination of both can also be done. Then the submit can be realized by a normal link that starts the submit. The different ways can be selected by changing the value of a hidden input field.

External Data

If pictures or other external static data from the server is needed, use a dynamically created URL instead of a static one. Use an environment variable, set by the Natural Web Server Extension to specify a non-standard path.

Setup your own variable for pictures in the Natural Web Server Extension initialization file:

```
SETENV=pictures:=/gif
```


Modular Pages

For a complex application, it is useful to define subroutines for special parts of the generated page.

- Page design split input: head, body, menu ...
- Use external subroutines.

Example of a modular generation page:

```
* Check given parameter of mandatory value is set.
IF #DEMO-PARM.#PERSON EQ " " THEN
  #DEMO-PARM.#MSG := 'Please Select a person!'
  INCLUDE D4-BACK
END-IF
*
* Specify the name of this page
#DEMO-PARM.#TODO := "SHOW"
#DEMO-PARM.#CAPTION := 'Show'
*
* Define the Navigationbar items to be shown
#DEMO-MENU.#ME := "T" /* Menu
#DEMO-MENU.#CH := "T" /* Change
#DEMO-MENU.#DE := "T" /* Delete
*
* Define the Navigationbar back to Select/Browse
PERFORM D4-BACKB #DEMO-MENU #DEMO-PARM
*
* Generate Start of HTML page and Navigationbar
PERFORM D4-START #DEMO-MENU #DEMO-PARM
*
* add global ID as hidden input field
PERFORM D4-HPERS #DEMO-MENU #DEMO-PARM
*
* Read Employee data from the Database
CALLNAT "D4EMSHOW" #DEMO-PARM.#PERSON #OUT-P-NAME
        #OUT-P-FIRST-NAME #OUT-P-BIRTH #OUT-P-SEX
        #OUT-P-DEPT #OUT-P-CITY #OUT-P-AREA-CODE
        #OUT-P-ZIP #OUT-P-COUNTRY #OUT-P-PHONE
        #OUT-P-PICTURE #DEMO-PARM.#MSG
*
* Generate HTML output for this page
INCLUDE D4-DISPL
*
* Generate end of HTML page
PERFORM D4-END #DEMO-MENU #DEMO-PARM
END-SUBROUTINE
```


Chosen Program Design

The application uses one startup page and one dispatch page. These pages contain a form which contains all necessary input fields. Data will be passed as hidden fields from one page to another. The different functions of the page will be selected by the value of a specific field. The value will be set via a JavaScript program used on this page. This makes it possible to create a cursor-sensitive selection menu.

The main programs called for every page are the generation of the page head, the menu and the page body. The page head and body contain the layout information for the table structure used on the page. Data given from the previous page will be read to a parameter data area that will be passed to all subroutines used. This parameter data area contains an array to set up the relevant parts of the menu.

The layout of the pages is encapsulated with copycode if the layout is used more than once. Access to the database is separated to separate subprograms.

External data as gifs and the JavaScript source can be reallocated using variables defined in the Natural Web Server Extension.

Functional Parts

Main Programs

D4ENTER, Subprogram
: entrance tunnel

D4MENU, Subprogram
: main menu

D4CHOOSE, Subprogram
: dispatcher program

Database Access

D4EMBROW Subroutine
: browse

D4EMCHAN Subprogram
: change

D4EMDEL, Subprogram
: delete

D4EMLIST, Subroutine
: list

D4EMNEW, Subprogram
: add new

D4EMSEL, Subroutine
: select

D4EMSHOW, Subprogram
: show

General HTML Layout

D4-BACK, Copycode
: relocate if nothing is selected

D4-BACKB, Subroutine
: back button to return correct to list/browse

D4-DISPL, Copycode
: Display Screen for Show/Delete

D4-END, Subroutine
: generate the "end" of a demo HTML page

D4-HGLOB, Subroutine
: write "global" variables as `START, FROM, TO, OT, STATUS` to the HTML page

D4-HPERS, Subroutine
: write the "global" variable Person "ID" to the HTML page

D4-IPAGE, Copycode
: add a person

D4-MENUB, Subroutine
: generate the menu bar for the demo HTML page

D4-MODIF, Copycode
: input screen for change/new

D4-MSG, Subroutine
: generate the message line

D4-PARM, Parameter Data Area
: cross application data

D4-RGLOB, Subroutine
: read variables send by the previous page

D4-START, Subroutine
: generate the "start" of a demo HTML page

Sub HTML-Pages

D4BROWSE, Subroutine
: browse persons

D4CHANGE, Subroutine
: change a person

D4DELETE, Subroutine
: delete a person

D4LIST, Subroutine
: list persons

D4NEW, Subroutine
: add a person

D4SELECT, Subroutine
: select a person

D4SHOW, Subroutine
: show a person

Special Purpose

D4NOTIMP, Text
: static page for not implemented

D4TEMPEL, Subprogram
: display static pages saved as Natural text members

